字符串相关算法介绍

罗雨屏

清华大学　交叉信息研究院

2014 年 10 月 2 日

KMP

- 用途：给定一个串 S, 询问 T 是否在 S 中出现过
- 暴力：$O(nm)$
  - `strstr` 就是这么写的
  - 需要优化
- 考虑这样一种情况
  - S 为 `abcabcabd`，T 为 `abcabd`
  - 已经成功匹配了前五个字母 `abcab`，在 `d` 处失败了

KMP

- 用途：给定一个串 S, 询问 T 是否在 S 中出现过
- 暴力：$O(nm)$
  - `strstr` 就是这么写的
  - 需要优化
- 考虑这样一种情况
  - S 为 `abcabcabd`，T 为 `abcabd`
  - 已经成功匹配了前五个字母 `abcab`，在 `d` 处失败了
- 已经有的信息
  - 我们已经知道了接下来若干个字母的信息，为什么要重新开始做匹配呢？
  - 即，如果暴力试下一个子串，则一定是 `bcab` 开头的，为什么还要试下去？
  - 再下一个 `cab`：也可忽略
  - 再下一个 `ab`：一定可以匹配两个字符！

KMP cont'd

- 对于 T 的每一个前缀，预处理出在这个地方匹配“失败”时，可以忽略几个字符
  - 一个最长的后缀，使得其为 T 的一个前缀
  - 定义 f_i 表示这个后缀的长度

KMP cont'd

- 对于 T 的每一个前缀，预处理出在这个地方匹配“失败”时，可以忽略几个字符
  - 一个最长的后缀，使得其为 T 的一个前缀
  - 定义 f_i 表示这个后缀的长度
- 求 f_i ：递推

Theorem (KMP)

$$f_{n+1} \leq f_n + 1.$$

- 何时等于？不等于怎么办？

exKMP

- 对于 S 的每个后缀，求其与 S 的 LCP
- 令 f_i 表示 S 的第 i 个后缀与 S 的 LCP
  - 维护 $\max(i+f_i)$，令此时的 i 为 t
- 若已知 $f_{1,\ldots,k-1}$ 想求 f_k
  - 利用 t 可以得到 k 的一个下界
  - 剩下的暴力
- 复杂度：对于每个 k，每暴力一个字符，$\max(i+f_i)$ 就会增加 1
  - 有上界 n
  - 故复杂度 $O(n)$

Manacher

- 求 S 的每个极长奇回文串的长度
- 与 exKMP 很类似
  - 也是维护 $\max(i + f_i)$
- 复杂度同理，$O(n)$

Manacher

- 求 S 的每个极长奇回文串的长度
- 与 exKMP 很类似
  - 也是维护 $\max(i + f_i)$
- 复杂度同理，$O(n)$
- 偶数长度的回文串：加一位

Hash

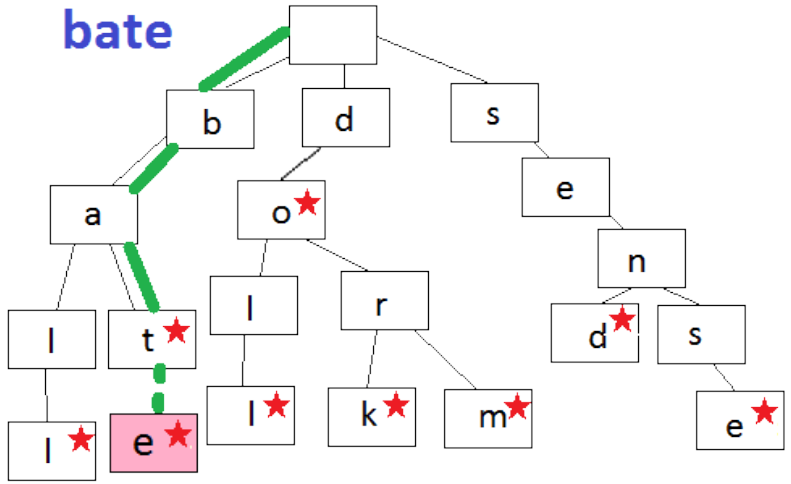
- 看成 p 进制数，再 $\mod q$
- p, q 如何选取
  - p 质数，较大即可
  - q 可以方便的取 2^{64} ，否则质数效果比较好
- 如何求子串 Hash
  - 规定 S 的 Hash 为 $\sum_{i=1}^{n} S_i p^{-i}$
  - 预处理：求前缀和
  - 查询：求区间和，再乘上 p^s
- 卡 Hash 的方法

最小表示法

- 暴力的一种优化
- 考虑从 s 开始和从 t 开始的两个后缀
- 若 $S_{s+} < S_{t+}$ 则 $S_{t,\ldots,t+x}$ 不可能为最优解，其中 $x = LCP(S_i, S_j)$
- 观察得知，每暴力求 $O(k)$ 的 LCP ，就可以使得答案范围缩小 $O(k)$
- 复杂度线性

Trie

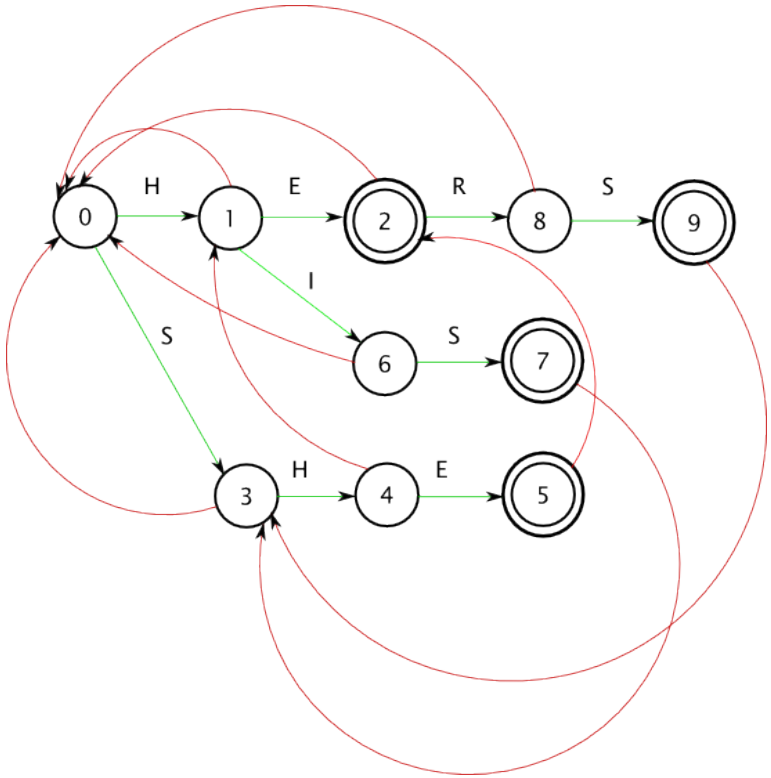
- 常说的字母树



- 实际中信息是存在边上的
- 一点小优化：如果一个节点只有一个孩子，那么可以把这个节点省略掉
  - Compacted-trie
  - 空间复杂度降为 $O(n)$，n 为串数目

AC automaton

- Aho-Corasick automaton
  - KMP 在树上的加强版



AC automaton cont'd

- 每个点的 fail 指针意义与 KMP 的 next 类似
  - 令 S 为根节点到当前节点的串，则 $fail$ 指针指向的串为 **最长的/存在于 Trie 上的串/使得这个串是 S 的前缀/且不是 S 本身/的串**
  - 求法也和 KMP 类似
- 线性构造

Trie 图

- AC automaton 的加强版
- 对于 Trie 上的每个节点，求出这个串添加一个字符 c 后会走到哪个节点去
- 类似于 KMP 的构建不知道复杂度

后缀数组 SA

- 给定一个串 S，要求给 S 的每个后缀排序
- 倍增 $O(n \log n)$
  - 令 $SA^{(k)}$ 表示比较前 2^k 个字符的后缀数组
  - 已知 $SA^{(k)}$ 可以在 $O(n)$ 的时间内求出 $SA^{(k+1)}$
  - 倍增 $O(\log n)$ 次即可

后缀数组 SA

- 重大武器 $height$ 数组
  - $height_i$ 表示 SA_i 与 SA_{i-1} 的 LCP 长度
- 作用：查询后缀的 LCP $\Longleftrightarrow$ RMQ
  - 想怎么鞭尸就怎么鞭尸
- 令 $rank_i$ 表示 i 在 SA 中的位置
- 注意到

$$height_{rank_i} - 1 \leq height_{rank_{i+1}}$$

  - 按照 $rank$ 的大小来求 $height$，可以做到线性

简单计数问题

- 给定一个串 S
- 求有多少个本质不同的子串

简单计数问题

- 总串数目减去 height 之和
- 请同学列举一下 abcabd 的所有子串

SPOJ SUBLEX

- 给定一个串，求其第 k 字典序小的子串
- T 组询问
-
- $|S| \leq 10^5, T \leq 10^5$

SPOJ SUBLEX

- 按照上个题的思路继续做下去
- 知道了一个后缀对应了几个本质不同的子串
  - 而且还是按照字典序排列的
- 直接二分得出第 k 个子串属于第一个后缀，然后直接计数即可
  - $O(\log n)$ 每次回答

JSOI

- 给定一个数字串 S 以及 n 个串 $T_{1,\ldots,n}$
- 求 T_i 在 S 中出现的次数
- 将 T_i 的每个字符加上上次答案模 10 ：强制在线
-
- $|S| \leq 10^6, \sum |T_i| \leq 10^5$

JSOI

- 现在唯一可以用来做字符串计数的武器：AC 自动机
- AC 自动机需要对 T 建
  - 强制在线

JSOI

- 现在唯一可以用来做字符串计数的武器：AC 自动机
- AC 自动机需要对 T 建
  - 强制在线
- 所有可能的询问串只有 10 种
  - 将询问范围扩大 10 倍 $\rightarrow$ 允许离线

SPOJ LCS2

- 求多个串的 LCS
  - 最长公共子串
- $n \leq 10^4, \sum |S_i| \leq 10^5$

SPOJ LCS2

- 首先把所有串拼在一起，任意两个串之间用一个特殊字符隔开
- 考虑所有以 LCS 为前缀的后缀组成的区间
  - 对于所有 S_i ，一定有一个后缀在里面

SPOJ LCS2

- 首先把所有串拼在一起，任意两个串之间用一个特殊字符隔开
- 考虑所有以 LCS 为前缀的后缀组成的区间
  - 对于所有 S_i ，一定有一个后缀在里面
- 二分答案 ans
  - 有很多被小于 ans 的 $height$ 隔开的区间
  - 如果一个区间内对于每个 S_i 都存在一个其的后缀，则当前答案是可行的
  - 复杂度 $O(n \log n)$

SPOJ LCS2

- 首先把所有串拼在一起，任意两个串之间用一个特殊字符隔开
- 考虑所有以 LCS 为前缀的后缀组成的区间
  - 对于所有 S_i ，一定有一个后缀在里面
- 二分答案 ans
  - 有很多被小于 ans 的 $height$ 隔开的区间
  - 如果一个区间内对于每个 S_i 都存在一个其的后缀，则当前答案是可行的
  - 复杂度 $O(n \log n)$
- 思考
  - 可以不二分答案吗

CERC 2008

- 给定后缀数组 SA
- 求一个满足条件的原串
- 字符集 26
- 保证存在一组合法解
-
- $n \leq 5 \times 10^5$

CERC 2008

- 令 SA 为后缀数组，$rank$ 为排名数组，S 为原串
- 将所有后缀依次写下来，我们的目标是 ~~没有蛀牙~~ 尽量减少 S 字符集大小，也就是尽量使得 SA 中连续两个字符相同
- 若 $rank_{SA_i+1} > rank_{SA_{i+1}+1}$ 则必有 $S_{SA_i} < S_{SA_{i+1}}$
- 否则我们可以令 $S_{SA_i} = S_{SA_{i+1}}$ ，肯定是可以满足要求的

盾盾的打字机

- 给定一个 d/r 串，每次可以删除一个长度为偶数的回文串的后面一部分
- 求删除后的串的最短长度
- $n \leq 10^7$

盾盾的打字机

- 如果出现了 drr 或 rdd，则后面的字符一定都可以被消除
  - 先递归处理出后面的
    - 不管怎么处理，只要不可继续操作，一定是 d/r 交替
  - 如果当前是 drr 接下来是 rd，则 drrrd $\rightarrow$ drrd $\rightarrow$ dr
  - 如果接下来是 dr 则 drrdr to drr
  - 只剩一个 d 则 drrd $\rightarrow$ dr
- 如果没有出现 drr 或 drd
  - 必定形如 ddddrdrdrd
  - 找到剩下的串中 d/r 交替出现的次数即可

最长重复子串

- 给定一个串 S，求一个最长的 T 满足 TT 在 S 中出现过
- $|S| \leq 10^5$

最长重复子串

- 枚举 T 的长度 L，每次查询是否存在满足条件的 T
- 把序列分成若干段，每一段长度为 L
- 如果存在的话必定存在两个相邻的分界点被包括在 TT 内部，即形如 ABAB 其中 AB 被隔开了
- 对于相邻两个分界点来说，求正向 LCP 长度为 x，逆向 LCP 长度为 y
- 有 $|B| \leq x, |A| \leq y$ 推出 $x + y \geq n$
  - 而且这也是存在 AB 的充要条件
- 对于 L 处理时间为 $O(\frac{n}{L} \log n)$，总时间复杂度为 $O(n \log^2 n)$

A Horrible Poem

- 给定一个长度为 n 的字符串 S
- 有若干个询问，每次询问一个子串的最短循环节长度
- 串 T 是 S 的循环节，当且仅当一个整数 k 使得 $T^k = S$
- $|S| \leq 10^6, Q \leq 10^4$

A Horrible Poem

- 注意到，如果 T 是 S 的循环节，则有 $|T|||S|$
- 如何求一个串的最短循环节？
  - KMP：线性，太慢
  - 枚举循环节长度，直接用 Hash 判断：$O(\sqrt{n})$
  - 仍然太慢
- 注意到如果 $|T_1| = a$ 是 S 的循环节，$|T_2| = b$ 是 S 的循环节，那么 $\gcd(a, b)$ 也是 S 的循环节
- 对于每组询问的区间长度 t，我们枚举 t 的每个质因数 p，找一个最大的 k 满足 $\frac{n}{p^k}$ 是循环节
- 没必要用 CRT 合并，因为实质上这是在求最短循环节的每个质因数的指数

The Shortest Period

- 给定一个串 S，要求删掉至多一个字符，使得最小循环串长度最小
- X 是 Y 的循环串，当且仅当 Y 是 X^{∞} 的前缀
- $|S| \leq 10^5$

The Shortest Period

- 考虑枚举答案 ans
- 如果删除的位置在前 ans 个字节中，则
  - S 除去前 $ans+1$ 个字符后得到的串存在一个长度为 ans 的循环串
    - 用 Hash 检验一下即可
  - S 的前 $ans+1$ 个字符中删去一个字符后能得到上述长度为 anS 的循环串
    - 求 LCP 后省略一个字符再用 Hash 判断是否可行
- 如果不在前 ans 个字符中，则
  - 可以求出删除的位置。如果这个字符和循环串对应位置相同，则我们不可能删除它
  - 所以删除的位置必定为对应不起来的位置：S 和 S_{ans+} 比较时第一个不同的字符
  - 再次判断是否可行即可
- 每次枚举的时间复杂度是 $O(\log n)$，总体时间复杂度 $O(n \log n)$

prefixuffix

- 给定一个串 S
- 求一个最大的 L 满足：S_{L-} 与 S_{-L+} 循环同构，且两个串不重叠
- $|S| \leq 10^6$

prefixuffix

- 两个串循环同构必为 AB 和 BA 这种形式
- 令 f_i 表示当 $|A| = i$ 时 B 的最大长度
- 有 $S_{i+1,i+f_i} = S_{-(i+f_i),-(i+1)}$ ，故 $S_{i+2,i+f_i-1} = S_{-(i+f_i-1),-(i+2)}$
- 推出

$$f_i + 2 \leq f_{i+1}$$

- 按照 i 从大到小求 f_i